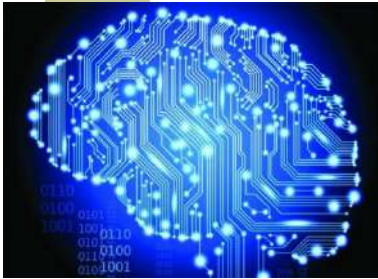




# Andmebaasid

Erki Eessaar



# Andmed

- ◆ Andmed on **märkide hulgad**, mis on mingis *keskkonnas* salvestatud.
- ◆ Andmed võivad olla näiteks:
  - märgid *savitahvlil*,
  - numbrid või tekst *paberil*,
  - bitid ja baidid *arvuti mälus*,
  - kujutised *filmilindil*,
  - meeldejäetud faktid *inimese ajus*, mis on seal salvestatud elektrokeemiliste impulssidena.

# Andmekandjate näiteid



Alternatiiviks regulaarne andmete ülekanne uue tehnoloogiaga kandjatele.

# Digitaalsete andmete pikaajaline säilitamine

- ◆ Selleks, et digitaalselt talletatud andmeid oleks võimalik ka tulevikus (näiteks 100 või 1000 aasta pärast) kasutada, tuleb lisaks andmekandjatele säilitada ka täpne info, kuidas ehitada masinad ja tarkvara, mis suudab sellistelt andmekandjalt infot lugeda.
  - Gosh, P., 2015. Google's Vint Cerf warns of 'digital Dark Age'. BBC, 13.02.2015 [WWW]  
<https://www.bbc.com/news/science-environment-31450389>

# Informatsioon

- ◆ Informatsioon on andmetega süstemaatiliselt tehtud *operatsioonide* (andmetöötuse) tulemus, mis on saaja poolt *tõlgendatav* ja *mõistetav* ning aitab tal midagi:

- järeldada,
- soovitada,
- otsustada,
- kinnitada,
- kokku võtta.

*Our World in Data*

<https://ourworldindata.org/> ja

*Population Pyramid*

<https://www.populationpyramid.net/>

Visualiseeritakse andmeid maailma rahvastiku ja elustandardi hetkeseisu ja muutuste kohta.

# Informatsioon (2)

## ◆ Informatsioon võib olla

- aruanne,

- analüüs,

- andmed, mida on korrastatud

- andmetabeli read ja veerud on mingi reegli järgi sorteeritud, et andmetes peituvaid seoseid oleks visuaalsel (paremini) näha

- sõnaline vastus,

- joonis, ...

<https://www.statista.com/chartoftheday/>  
Visuaalsed kokkuvõtted (aruanded) väga paljude erinevate andemhulkade põhjal.



# Hägune piir andmete ja informatsiooni vahel

- ◆ Samas kasutavad mitmed autorid termineid *andmed* ja *informatsioon* sünonüümidenä.
  - Andmetöötlus võib koosneda mitmest sammust.
  - Ühe andmetöötlusoperatsiooni tulemuse leitud informatsioon võib olla teise operatsiooni lähteandmeteks.

# Andmebaas

- ♦ Andmebaas on hulk *tõeseid väiteid* (fakte) reaalse maailma kohta, mis on salvestatud mingis keskkonnas.
- ♦ Andmebaas peab võimaldama vastata päringutele praegu ja tulevikus.
- ♦ Selleks, et vajalikke andmeid leida, peavad andmed olema organiseeritud (korrastatud) mingi struktuuri järgi.



Andmebaas **ei** pea olema  
loodud arvutisüsteemi  
kasutades

# Andmete talletamise ajalugu



# Andmete talletamise ajalugu (2)



- ◆ Inkad kasutasid arvepidamiseks ja andmevahetuseks *khipusid* e rääkivaid sõlmi.
- ◆ Koosnevad nööridest, milles on sõlmed.
- ◆ Sõlmede abil esitatakse arve.
- ◆ <https://khipukamayuq.fas.harvard.edu/>

# Andmed IT-süsteemides

- ◆ Arvutisüsteemi mõttes on andmed osa tarkvarast.
- ◆ Arvutisüsteemi osad.
  - Riistvara.
    - *Käegakatsutav* osa süsteemist – nt arvuti kast.
  - Tarkavara.
    - *Mitte käegakatsutav* osa süsteemist.
    - Programmid e käskude jadad andmete töötlemiseks.
    - Andmed e materjalid, millega programmid töötavad.

# Andmed IT-süsteemides (2)



- ◆ Andmed peavad olema õigel ajal, õigel kujul, õigetele soovijatele kättesaadavad.
- ◆ Andmeid võib hoida spetsiaalse tarkvara – **andmebaasisüsteemi** – abil loodud andmebaasis.

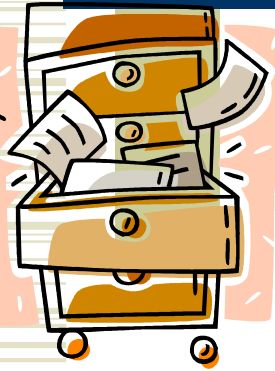
# Andmebaasisüsteem (*Database Management System, DBMS*)



<https://db-engines.com/en/ranking>

- ◆ Tarkvara, mis:
  - võimaldab andmebaase luua ja hallata ja
  - on nagu *müür* mille sees on andmebaasid (nagu aiad) ja värav/väravavaht, läbi mille peab toimuma igasugune nende andmebaaside kasutus.





# Andmebaasisüsteemide eelne ajastu

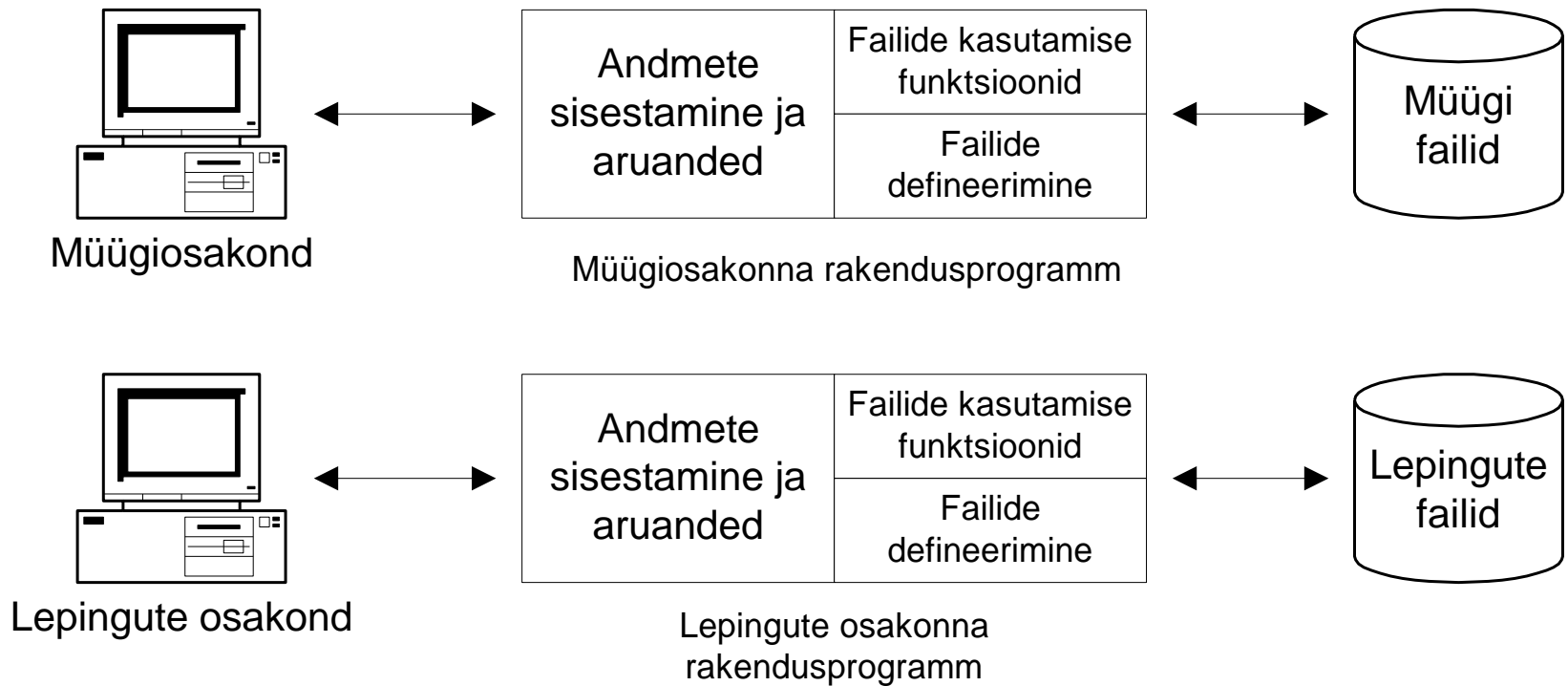
- ◆ Andmeid võib kirjutada *paberile*.



# Andmebaasisüsteemide eelne ajastu (2)

- ◆ Andmeid võib hoida arvutis failidena nii, et andmeid kasutavad programmid pöörduvad *otse failide poole*.

# Failipõhine süsteem





# Ülesande püstitus

- ♦ Leia *erinevad* töötajate perenimed.

# Failipõhise lahenduse *eeldused*

- ◆ Olemas fail nimega *tootaja.txt* ja programmil õigus seda lugeda.
- ◆ Andmete puhul on kasutusel tekstiline kodeering.
- ◆ Failis on ainult perenimed.
- ◆ Kirjete eraldajaks on reavahetus.
  - Viimase kirje järel reavahetus.
- ◆ Perenimed on tähestiku järgi sorteeritud.
  - Kui see muutub, siis programm töötab, kuid tulemus vale!
- ◆ Ühes reas oleva stringi pikkus ei ole suurem kui  $1024-1=1023$  baiti.

# Ülesande failipõhine lahendus

```
<?php
$seelmine="";
/*Faili avamine lugemiseks*/
$handle = @fopen("tootaja.txt", "r");
if ($handle) {
    while (!feof($handle)) { /*Tsükel kuni on veel kirjeid*/
        $buffer = fgets($handle, 1024); /*Kirje lugemine failist*/
        if ($buffer!=$seelmine) {
            $seelmine = $buffer;
            echo nl2br($buffer);
        }
    }
    /*Faili sulgemine*/
    fclose($handle); }
?>
```

Kask  
Metsis  
Saabas  
Sukk



Kask  
Kask  
Metsis  
Saabas  
Sukk  
Sukk

Programmi kirjutaja *realiseerib*  
**korduste eemaldamise**  
**algoritmi.**

Kui nt failis pole kirjed  
sorteeritud, siis annab programm  
vale tulemuse.

# Lahendus

## SQL-andmebaasisüsteemis

- ◆ Eeldused.
  - Andmebaasis on tabel *Tootaja*, kus on veerg *perenimi* (võib ka olla teisi veerge).
  - Kasutajal on õigus lugeda tabelis *Tootaja* veerus *perenimi* olevaid andmeid.
- ◆ Lahenduseks olev päring (**SQL** keeles).
  - **SELECT DISTINCT** perenimi FROM Tootaja;
    - Päringu kirjutaja **deklareerib**, et soovib tulemusest korduvate ridade eemaldamist ning andmebaasisüsteem valib parima viisi selle saavutamiseks.

# SQL andmekäitluskeele lausete töötlemine

```
SELECT DISTINCT perenimi FROM Tootaja;
```



SQL SELECT lause – *mida* tuleb leida  
(**deklaratiivne** lause)

```
HashAggregate (cost=10.62..11.12 rows=50 width=516)
  Group Key: perenimi
  -> Seq Scan on tootaja (cost=0.00..10.50 rows=50 width=516)
```



Täitmisplaan – **imperatiivne** programm lause täitmiseks – *kuidas* andmed leida. Koostatakse **andmebaasisüsteemi** poolt automaatselt

perenimi
Kadakas
Lehis
Kask

Lause täitmise  
tulemus

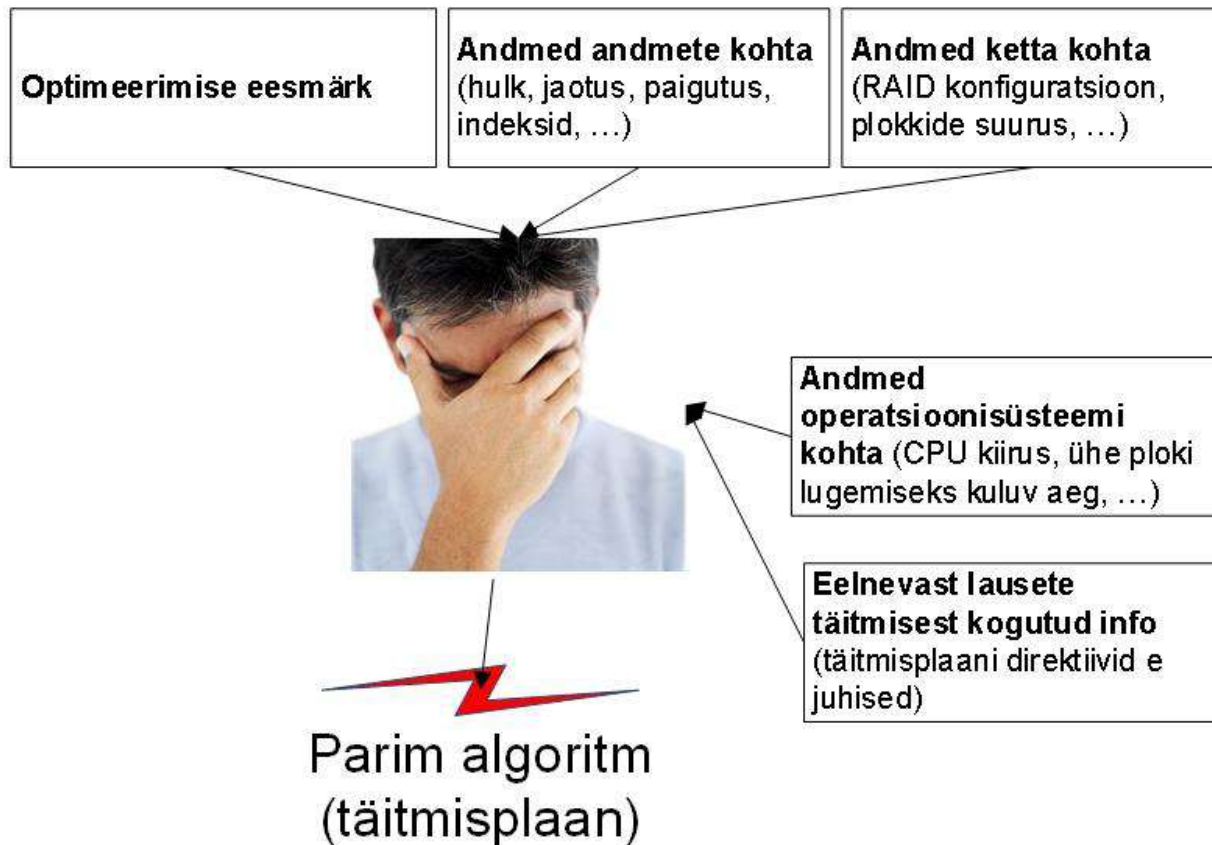
# Imperatiivsus vs. deklarativsus

- ♦ Failipõhise lahenduse näide on **imperatiivne** programm, mis kujutab endast *käskude järjendit*.
  - Programmi täitmisel toimub käskude täitmine etteantud järjekorras.
- ♦ Andmebaasikeelte (nt SQL) lause näide on **deklarativne**.
  - Päringu koostaja ütles süsteemile, *milliseid* andmeid leida, kuid mitte *kuidas* seda teha.

# Deklaratiivse andmebaasikeele kasutamise eelis

- ◆ Andmebaasisüsteem valib ise (võttes muuhulgas arvesse andmebaasis olevaid andmeid), milliseid operatsioone kasutades (**kuidas**) on kõige parem lause täita.
  - Andmebaasisüsteemi käsutuses peab olema **statistika** andmebaasis olevate andmete kohta
    - Näiteks kui palju ridu on tabelis või kui palju erinevaid väärtuseid on veerus.
- ◆ Andmebaasi kasutajate *produktiivsus suureneb!*

# Deklaratiivse andmebaasikeele kasutamise eelis (2)



Kui kasutate süsteemi, kus andmete leidmiseks/ muutmiseks tuleb kirjutada **imperatiivne** programm, siis peate ise leidma parima algoritmi (täitmisplaani), kuidas seda teha.



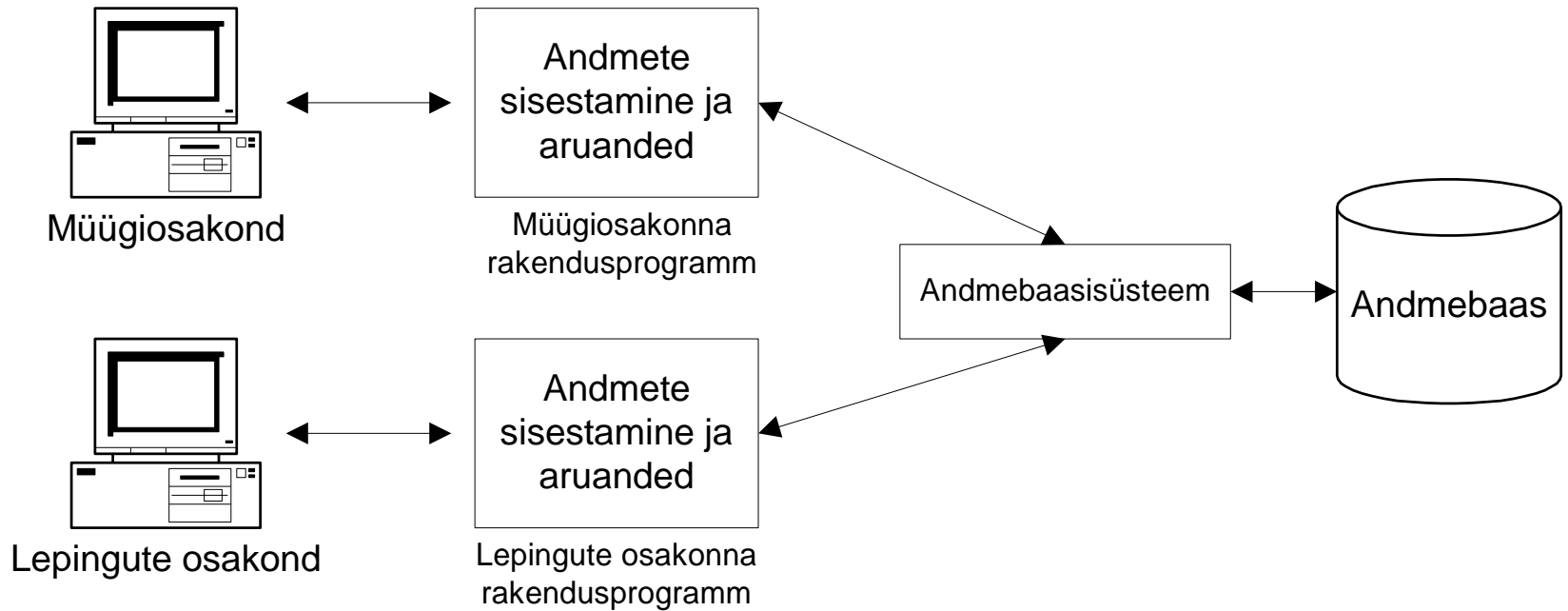
# Otsesest failide kasutamisest tulenevad puudused

- ◆ Andmete isoleeritus.
- ◆ Andmete sõltuvus neid kasutavast programmist.
  - Näide: andmete vastavust reeglitele kontrollib ainult rakendusprogramm.
- ◆ Programmide sõltuvus nende kasutatavatest andmetest (täpsemalt nende *füüsilisest* struktuurist). Näited.
  - Programm peab teadma indeksi olemasolust.
  - Väljade järjekord kirjes võib olla oluline.
  - Kirjete järjekord failis võib olla oluline.

# Otsest failide kasutamisest tulenevad puudused (2)

- ◆ Erinevate programmide andmete kodeeringu ja skeemi ühildamatus.
- ◆ Andmeid kasutav programm peab kõik failid ise avama ja sulgema.
- ◆ Võimalikud vaid fikseeritud päringud.
- ◆ Sageli ei ole mõeldud turvaprobleemidele, faili taastamisele.
  - Andmetele pääseb ligi "otse", faili poole pöördudes.
- ◆ Sageli võib faili korraga muuta vaid üks kasutaja.

# Andmebaasisüsteemi kasutav süsteem



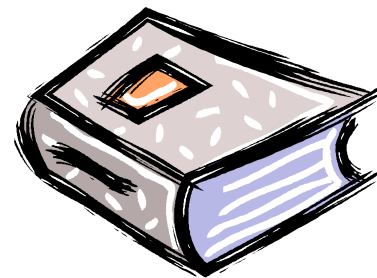
# Andmebaasisüsteem

## (andmebaasihaldur, andmebaasihalduse süsteem)

- ♦ *Andmebaasihaldur* on (Eesti standard EVS-ISO/IEC 2382. Infotehnoloogia sõnastik põhjal) riistvaral ja tarkvaral põhinev süsteem andmebaaside defineerimiseks, loomiseks, manipuleerimiseks, juhtimiseks, haldamiseks ja kasutamiseks.
- ♦ Andmebaasisüsteem on *tarkvarasüsteem*, mis võimaldab kasutajatel andmebaasi luua, kasutada, uuendada, hooldada ning sellele juurdepääsu kontrollida.

# Andmebaasisüsteem vs. andmebaas

- ◆ andmebaasisüsteem <> andmebaas
  - Täpselt samuti ei saa me öelda, et kirjutusmasin (vahend kirjatöö loomiseks) on sama, mis romaan (loometöö tulemus).



# Andmebaasisüsteem vs. andmebaas (2)

- ♦ Oracle  
andmebaasisüsteemi ametlik nimi on Oracle Database (Oracle Andmebaas).
- ♦ Ka PostgreSQL koduleht peab PostgreSQLi andmebaasiks.



ORACLE

Database /

Oracle Database 12c

Tegemist on andmebaasisüsteemide e andmebaasihalduritega – *Database Management System* (DBMS)

PostgreSQL: The World's Most Advanced Open Source Relational Database



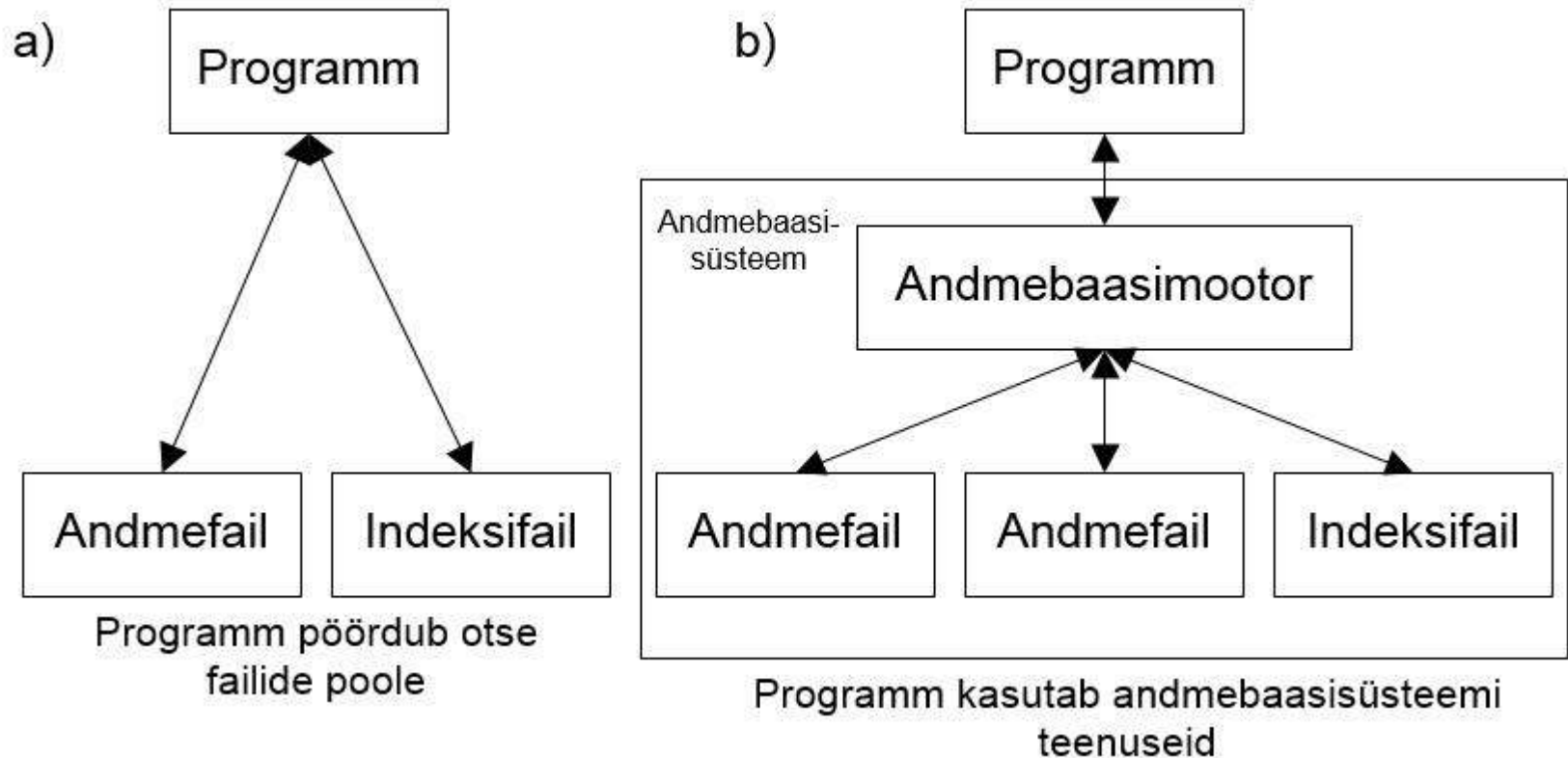
# Kuidas kõlab?

- ◆ Lõin PostgreSQL andmebaasi kasutades uue andmebaasi.

vs.

- ◆ Lõin PostgreSQL andmebaasisüsteemi kasutades uue andmebaasi.

# Kasutaja sõltumatus andmete füüsilisest salvestamisest





# Failipõhine süsteem vs. andmebaasisüsteem

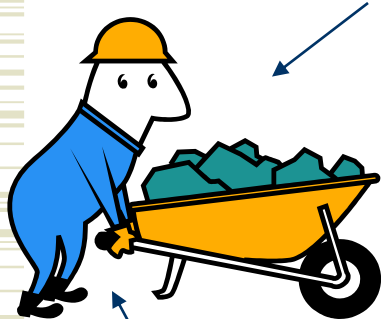
- ◆ Ka andmebaasisüsteemid salvestavad andmed füüsiliselt failidesse.
- ◆ Kuid erinevalt failipõhistest süsteemidest pakuvad andmebaasisüsteemid andmete kasutajatele "*vahekihi*", mis peidab (*kahjuks enamasti mitte täiesti*) nende eest andmete füüsilise salvestamisega seotud probleemid.
  - [https://en.wikipedia.org/wiki/Leaky\\_abstraction](https://en.wikipedia.org/wiki/Leaky_abstraction)

# Andmebaasisüsteemi mõned olulised funktsionaalsused

- ◆ Andmebaasisüsteem peab võimaldama andmeid mingi struktuuri järgi organiseerida.
  - Vt andmemudel.
- ◆ Andmebaasisüsteem peab kasutamiseks pakkuma keele(d), mille abil on võimalik andmeid otsida ja uuendada ja andmebaasi struktuuri hallata.

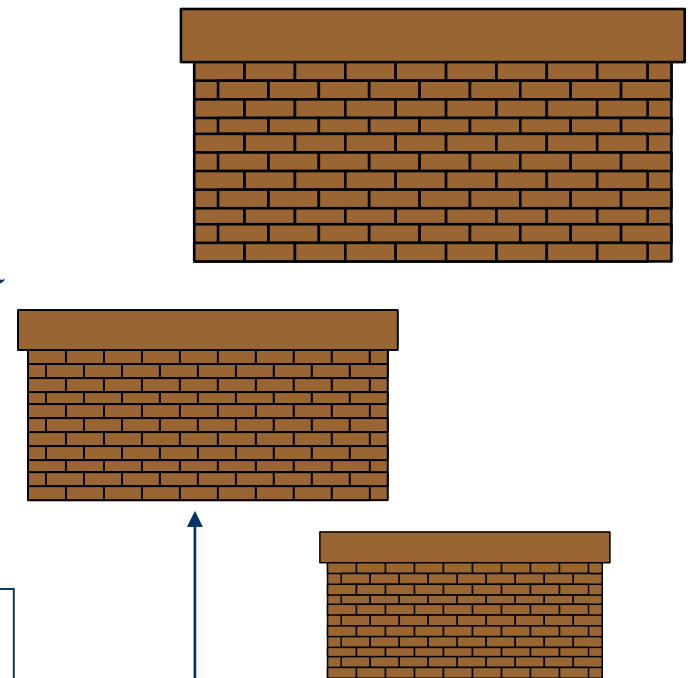
# Andmemudel (täendus 1)

Andmemudel määrab  
muuhulgas lubatud  
ehitusplokkide tüübid,  
mida kasutades saab luua  
palju erinevaid  
andmebaase



Andmebaasi  
arendaja

Näiteks SQL andmemudel  
määrab, et andmebaas peab  
koosnema tabelitest



Andmebaas

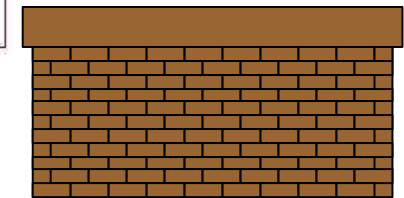
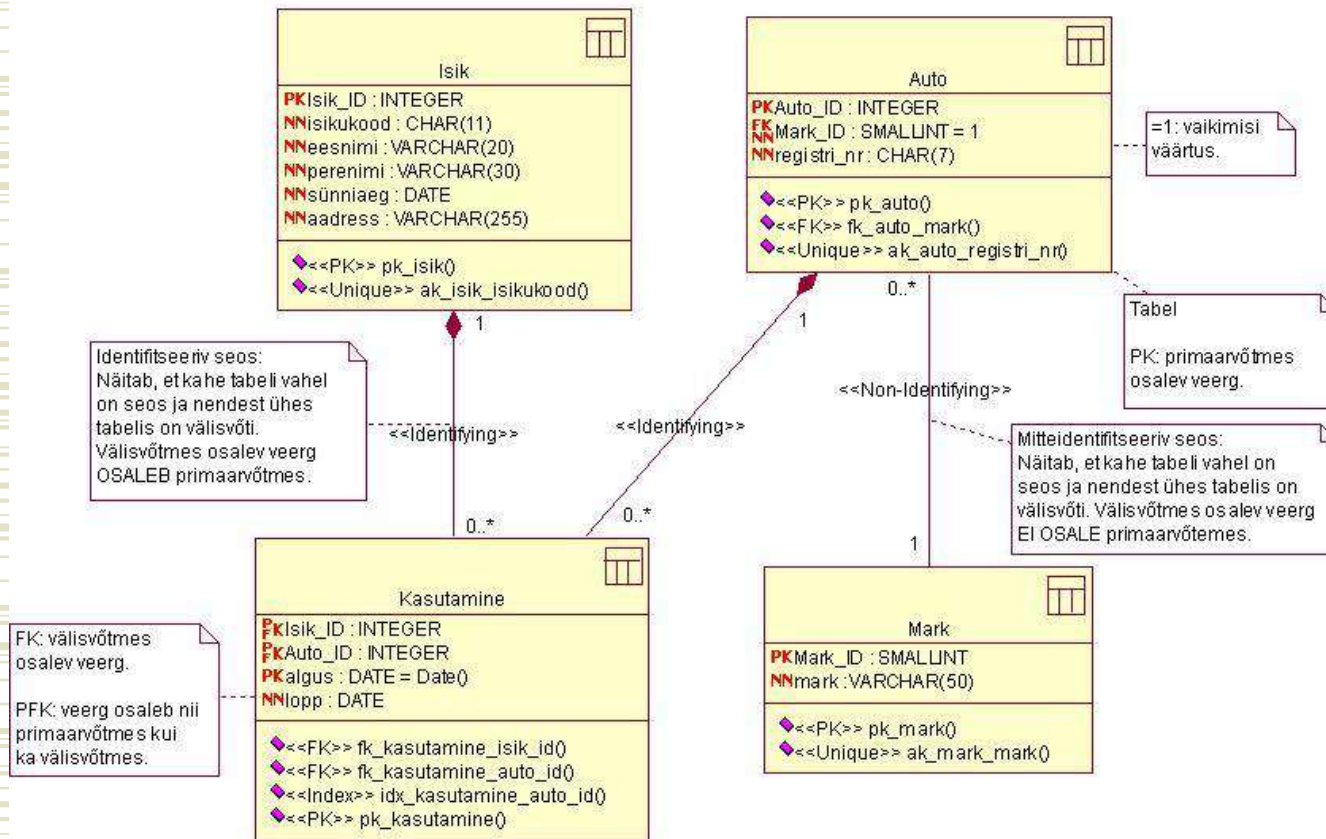
# Palju andmemudeleid (tähenduses 1)

- ◆ Hierarhiline.
  - Dokumendipõhine – XML või JSON dokumendid.
- ◆ Võrkstruktuur.
- ◆ **Relatsiooniline.**
  - **SQL:1992.**
- ◆ Objektorienteeritud.
- ◆ Objekt-relatsiooniline.
  - SQL:1999, SQL:2003, SQL:2006, SQL:2008, SQL:2011, SQL:2016
- ◆ Mitmemõõtmeline.
- ◆ Trans-relatsiooniline.
- ◆ Graafipõhised (omaduste graaf, RDF).

# Palju andmemudeleid (tähenduses 1) (2)

- ♦ <https://db-engines.com/en/ranking>  
(veerg Database Model)
  - *Relational DBMS* tähendab seal  
SQL-andmebaasisüsteemi (saab kasutada SQLi)
- ♦ *Multi-model* tähendab, et andmebaasisüsteem võimaldab andmeid organiseerida **erinevate** andmemudelite alusel.
- ♦ SQL andmebaasikeel ja selle aluseks olev andmemudel on **populaarsed**, kuid **pole ainsad**.

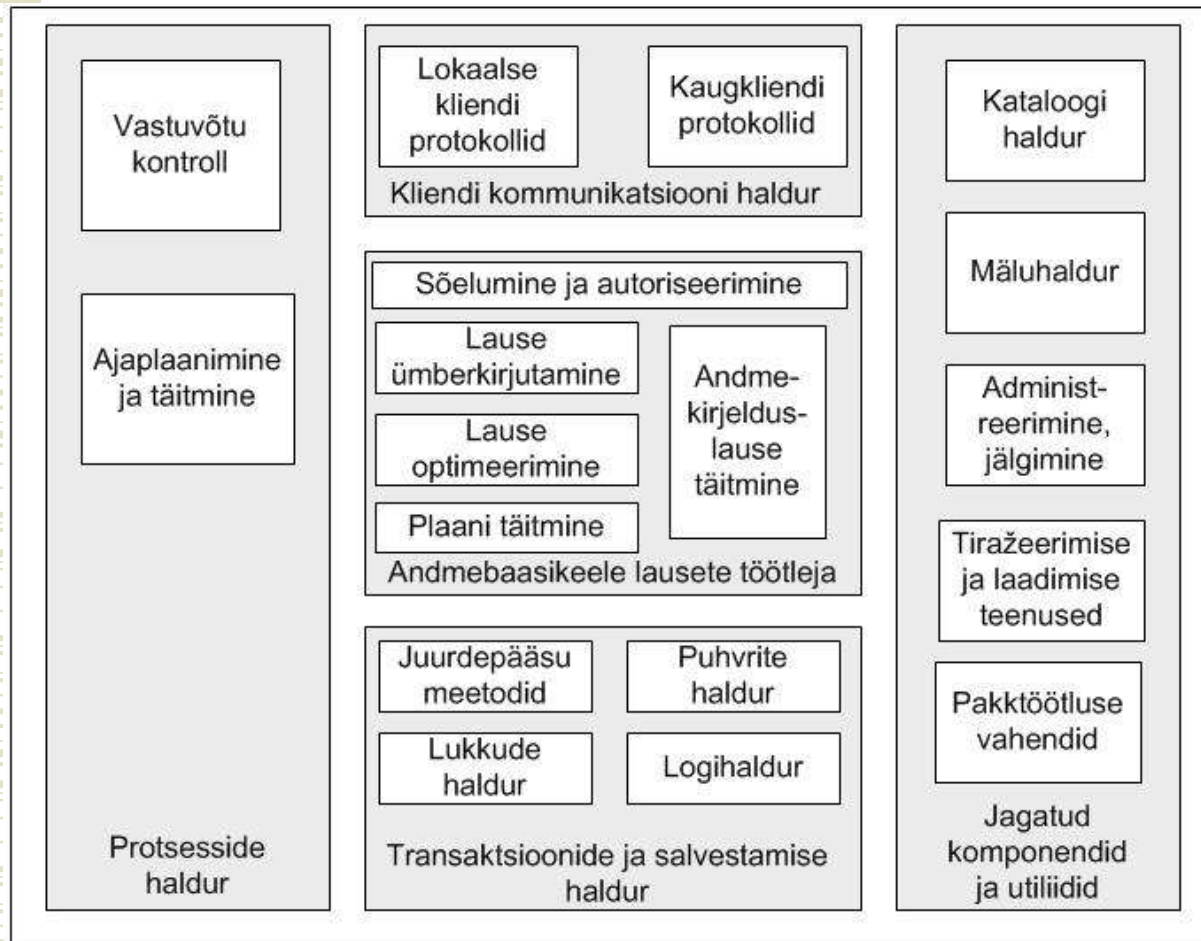
# Andmemudel (täendus 2)



Andmebaas

Konkreetsed andmebaasi kavand –  
füüsilise disaini andmemudel

# Hüpoteetiline andmebaasisüsteemi arhitektuur



Keeruline ja mahukas tarkvara.

Näiteks:

- PostgreSQL umbes **1.3 miljonit** koodirida,
- Oracle 12.2 peaaegu **25 miljonit** (C koodi) rida.



# Andmebaasisüsteemi analoogia – pank



**Klienditeenindus.** Suhtlus klientidega mingi protokollil alusel ning mingit keelt kasutades. Neilt nõuete vastuvõtt ja tulemuse tagastamine.

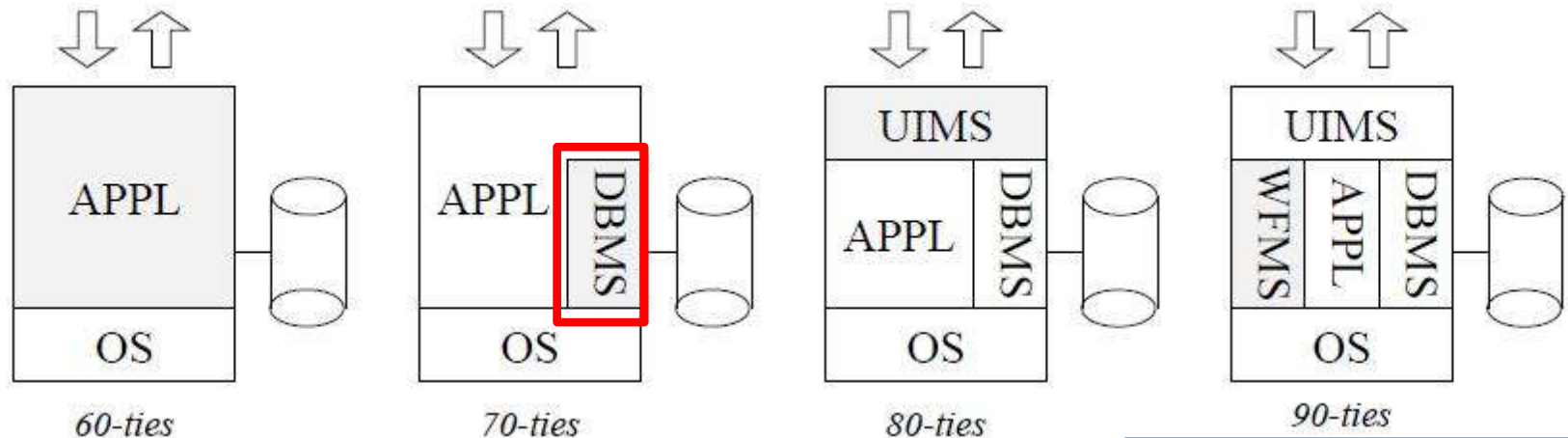
12.11.2020



**Tagakontor.** Klientide nõuete täitmine. Lisaks taustatöö (raamatupidamisest IT arenduseni), et panka töös hoida.



# Otstarbe lahususe (*separation of concerns*) areng rakendustes



APPL – rakendus

OS – operatsioonisüsteem

**DBMS – andmebaasisüsteem**

UIMS – kasutajaliidese halduse süsteem

WFMS – töövoogude haldamise süsteem

Aalst, W.M.P. van der (1996). Three good reasons for using a Petri-net-based workflow management system, In: Wakayama, T., Kannapan, S., Chan Meng Khoong, Navathe, S., Yates, J. (Eds.), *Proceedings of IPIC*, International Working Conference on Information and Process Integration in Enterprises (14-15 Nov. 1996, Cambridge, Massachusetts, USA), 179–201.

# Andmebaas ja infosüsteem

- ◆ Infosüsteem on ettevõtte info- ja süsteemitöö korralduse, meetodite ja vahendite kogusumma.
  - *Infotöö* käigus töödeldakse informatsiooni.
  - *Süsteemitöö* käigus arendatakse süsteemi.
  - Andmebaas(id) on infosüsteemi tuum.
- ◆ Andmebaas ja infosüsteem võivad eksisteerida ilma arvutisüsteemi toetuseta!

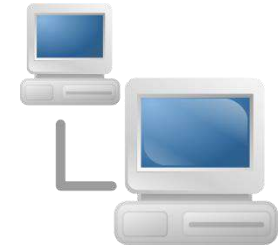
Slaid esitluses: Henno, J.  
(2012) *Emergence of  
Communication and  
Information*.  
Tallinn Technical  
University.

## Infosüsteem (2)

- ◆ K Hefner : kõik looduslikud süsteemid on **infotöötluste süsteemid**; iga selline süsteem saab infot vastu võtta, salvestada, töödelda ja edastada; võime infotöötlust läbi viia on kõigi süsteemide korral üks põhilisi võimeid; kogu universumist võib mõelda kui hiiglaslikust infotöötluste süsteemist.
  - K. Hefner (Ed.). *Evolution of Information Processing Systems*. Springer-Verlag 1992



## Infosüsteem (3)



- ◆ *Moodne* inimmaailmas kasutatav infosüsteem on **sotsiotehniline** süsteem – selles on:
  - sotsiaalne pool
    - inimesed,
    - organisatsioonid,
    - töökorraldus,
    - suhtlemine, motivatsioon ja **mõtteviis**
  - tehniline pool
    - info- ja süsteemitööd toetav riistvara ning tarkvara.



# Andmed ja andmebaas – piltlikult

- ◆ Andmed nagu **veri** (hapniku ja toitainete kandja) ettevõtte infosüsteemi vereringes.
- ◆ Andmebaas nagu **maks**.
  - Nii nagu maks talletab elutähtsaid aineid (sahhariide, rasvu, rauda ning vaske ja vitamiine), nii talletab andmebaas andmeid.
  - Nii nagu maks kahjutustab mürgiseid andmeid, nii aitavad andmebaasis jõustatud kitsendused tagada andmete terviklikkust.

# SQL-andmebaasi põhimõtteid

- ♦ Andmebaasis peavad andmed olema organiseeritud mingi struktuuri järgi.
- ♦ SQL pakub *ühe võimaliku* andmemudeli (täheenduses 1) kuidas sellises andmebaasis andmeid organiseerida.

# SQL-andmebaasi põhimõtteid (2)

- ◆ Andmebaasi loomiseks ja haldamiseks ning selles olevate andmete haldamiseks kasutatakse andmebaasikeelt SQL (*Structured Query Language*) kasutada võimaldavat andmebaasisüsteemi.
- ◆ SQL põhineb **relatsioonilisel** andmemudelil.
  - Üks võimalik (aga mitte parim võimalik) relatsioonilise mudeli **realisatsioon**.

# SQL-andmebaasi põhimõtteid (3)

- ◆ Kasutajale näib, et andmed on paigutatud **tabelitesse**.
- ◆ SQL-andmebaas on nime omav **nimega tabelite** ning nendega seotud teiste andmebaasiobjektide hulk (triggerid, arvujada generaatorid, domeenid, protseduurid jne).
- ◆ Igas tabelis on null või rohkem **rida** ja üks või rohkem **veergu**.



# Demonstratsioon

- ◆ PostgreSQL (13) põhjal.

# SQL standard

- ◆ Rahvusvaheline standard.
- ◆ Esimene versioon 1986.
- ◆ 2020. aasta sügisel kehtiv standardi versioon on **SQL:2016**.
- ◆ Standardis on hetkel **kümme** osa.
  - SQL:2011 – **4079** lehekülge.
  - <https://www.wiscorp.com/SQLStandards.html>

# SQL standard (2)

- ◆ Keele tuum on põhiliselt teises osas.
  - ISO/IEC 9075-2:yyyy Part 2: Foundation (SQL/Foundation) (SQL:2011 – **1483** lehekülge).
- ◆ SQL-andmebaasisüsteemid realiseerivad alamosa (mõned suuremad, teised väiksema) sellest standardist.
  - Igas süsteemis ka oma erisused.
  - Igas süsteemis oma SQLi dialekt e mägimurrak.

# SQL-andmebaas ei ole ainult tabelid



Baastabelite  
(tabelite; luuakse  
CREATE TABLE  
lausega) ümber võib  
SQL-andmebaasis  
olla rikkalik  
ökosüsteem erinevat  
tüüpi teistest  
andmebaasi-  
objektidest.

# SQL-andmebaas ei ole ainult tabelid (2)



- ◆ Kui **puu** = **baastabel**, siis ei ole SQL andmebaas, mitte palmide kasvandus, vaid näiteks vihmamets, kus puudega koos elab palju erinevaid tegelasi, kes kõik üksteist eluks vajavad.



# SQL-andmebaas ei ole ainult tabelid (3)

PostgreSQL

Oracle

- > Casts
- > Catalogs
- > Event Triggers
- > Extensions
- > Foreign Data Wrappers
- > Languages
- ▼ Schemas (1)
  - ▼ public
    - > Collations
    - > Domains
    - > FTS Configurations
    - > FTS Dictionaries
    - > FTS Parsers
    - > FTS Templates
    - > Foreign Tables
    - > Functions
    - > Materialized Views
    - > Procedures
    - > Sequences
    - > Tables
    - > Trigger Functions
    - > Types
    - > Views

- + Tables
- + Views
- + Indexes
- + Packages
- + Procedures
- + Functions
- + Operators
- + Queues
- + Queues Tables
- + Triggers
- + Types
- + Sequences
- + Materialized Views
- + Materialized View Logs
- + Synonyms
- + Public Synonyms
- + Database Links
- + Public Database Links
- + Directories
- + Editions
- + Application Express
- + Java
- + XML Schemas
- + XML DB Repository
- + OLAP Option
- + Scheduler
- + Recycle Bin
- + Other Users

# Palju erinevaid andmebaasitehnoloogiaid

- ◆ [https://www.csee.umbc.edu/~kalpakis/courses/661-fa15/papers/data\\_platform\\_map\\_june\\_2015-1.pdf](https://www.csee.umbc.edu/~kalpakis/courses/661-fa15/papers/data_platform_map_june_2015-1.pdf)

# Andmemahatude mõõtühikuid (SI-süsteem)

- ◆ 1 terabait (TB) = 1000 gigabaiti (GB)
- ◆ 1 petabait (PB) = 1000 terabaiti (TB)
- ◆ 1 eksabait (EB) = 1000 petabaiti (PB)
- ◆ 1 zettabait (ZB) = 1000 eksabaiti (EB)





# Näiteid andmemahutude muutusest

- ◆ 2012. aastal loodi ja tiražeeriti umbes **2.8 zettabaiti (2800 eksabaiti)** andmeid.
- ◆ Erinevad uuringud prognoosivad:
  - 2019. aastal luuakse ja tiražeeritakse umbes **30 zettabaiti (30000 eksabaiti)** andmeid.
  - 2020. aastal luuakse ja tiražeeritakse umbes **40 zettabaiti (40000 eksabaiti)** andmeid.
- ◆ Äriandmete maht kahekordistub iga 1.2 aastaga.

# Näiteid andmemahutudest



- ◆ Boeingu laineris tekib mootori töö **30 minutit** jälgimise tulemusena **10 terabaiti** jälgimisandmeid süsteemide töö kohta.
- ◆ Nelja mootoriga lainer tekitab ühe üle Atlandi lennuga **640 terabaiti** andmeid.
- ◆ Päevas on/oli maailmas (kui pole pandeemiat) vähemalt **25000 lendu**.

# Suurandmed (*big data*)



- ◆ Võib ettekujutada **tsunaamina**.
- ◆ Tsunaamit iseloomustab
  - kõike enda alla mattev materjali hulk,
  - pigem kasvav kui kahanev katkematu vool,
  - lisaks veele kannab see kaasa kõikvõimaliku suuruse, kujuga ning koostisega tükke.



# Suurandmete Moore'i seadus

- ◆ Andmete hulk kahekordistub iga kahe aastaga.
  - Moore'i seadus arvuti riistvara kohta: Mikrokiibil olevate transistoride arv kahekordistub iga kahe aasta järel.
- ◆ Mõlemad on **vaatlused!**

Idee tekkides pigem  
variant 1, nüüdseks  
variant 2.

# NoSQL

- ◆ Nimi võeti kasutusele 11.06.2009 ühel tarkvaraarendajate kohtumisel San Franciscos.
- ◆ Vaidlus, kas tähendab "No **to** SQL" või "Not **Only** SQL".
  - **Variant 2** paremini kooskõlas suundumusega *Polyglot Persistence* – isegi üks ja sama rakendus võib kasutada erinevatel andmemudelitel põhinevaid andmekogusid; erinevat tüüpi andmete jaoks sobivad erinevad andmemudelid ja süsteemid.



# Oht üle pingutada

## What the hell have you built.

- Did you just pick things at random?
- Why is Redis talking to MongoDB?
- Why do you even *use* MongoDB?

Goddamnit

Nevermind

FREE



Allikas:

<https://twitter.com/codinghorror/status/347070841059692545>

# NoSQL (Not Only SQL) andmebaasisüsteemid

- ◆ Andmehoidlate loomise süsteemid.
- ◆ Andmed võivad näiteks olla esitatud:
  - **võti-väärtus paaridena** (nt Redis),
  - **dokumentidena** (nt MongoDB),
  - **veergude perekondadena** (nt Cassandra),
  - **graafidena** (nt Neo4j).



<https://www.thoughtworks.com/insights/blog/nosql-databases-overview>

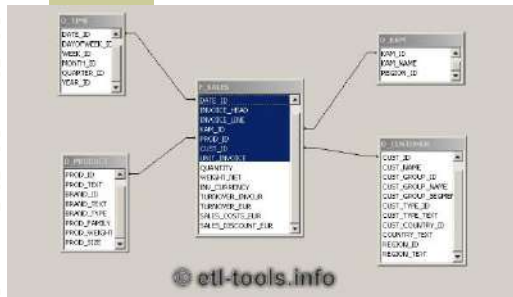
# NoSQL (Not Only SQL) andmebaasisüsteemid (2)

- ◆ Pakuvad "traditsiooniliste"  
SQL-andmebaasisüsteemidega (VanaSQL,  
OldSQL) võrreldes:
  - Paremat operatsioonide **töökiirust** väga **suurtel andmehulkadel**.
  - Paremat **skaleeritavust** (võimet kasvada järk-järgult).
    - Serverarvutite klastrisse e kobarasse saab lisada uusi arvuteid ja hakata andmeid ka sinna salvestama.



# Enamike NoSQL süsteemide ühised omadused

- ◆ Ei kasuta SQLi aluseks olevat andmemudelit.
- ◆ Töötavad hästi arvutite klastritel e kobaratel.
- ◆ Avatud lähtekoodiga.
- ◆ Kasvanud välja vajadusest pakkuda tuge 21. sajandi alguse suurtele veebipõhiste süsteemidele.
- ◆ "Skeemitud" andmed – skeem on kirjeldatud ilmutamata kujul rakenduste koodis.



# Skeemitusest

- ◆ Kui andmete lugeja ei tea andmete struktuuri (skeemi), siis pole tal võimalik andmetest aru saada.
- ◆ NoSQL süsteemides on kasutusel "skeem-lugemisel" lähenemine.
  - Andmete *kirjutamisel* **ei kontrollita** andmete struktuurile (skeemile) vastavust, kuid andmete *lugeja* **peab skeemi teadma**.

Aja jooksul  
süsteemid  
paranevad/  
täienevad.

# NoSQL süsteemide puudused

- ◆ Puudused võrreldes relatsiooniliste (ja SQL) andmebaasisüsteemidega:
  - selgelt defineeritud ning matemaatilise aluspõhjaga *andmemudeli* puudumine,
  - madalam *abstraktsioonitase*,
  - paljudes süsteemides kõrgtaseme deklaratiivse andmebaasikeele puudumine,
    - andmetöötluks tuleb kirjutada imperatiivset koodi
  - vähem ranged nõuded *transaktsioonidele*.

# Deklaratiivne vs. imperatiivne keel

```
var MJ = require("mongo-fast-join"),
    mongoJoin = new MJ();

/*
 * Say we have a collection of sales where each document holds a manual reference to the product sold. We can join the
 * full product document into each sale document. Lets also assume that each product has a reference to some
 * manufacturer info.
 */

mongoJoin
  .query(
    //say we have sales records and we store all the products for sale in a different collection
    db.collection("sales"),
    {}, //query statement
    {}, //fields
    {
      limit: 10000 //options
    }
  )
  .join({
    joinCollection: db.collection("products"),
    //respects the dot notation, multiple keys can be specified in this array
    leftKeys: ["product_id"],
    //This is the key of the document in the right hand document
    rightKeys: ["_id"],
    //This is the new subdocument that will be added to the result document
    newKey: "product"
  })
  .join({
    //say that we want to get the users that commented too
    joinCollection: db.collection("manufacturers"),
    //This is cool, you can join on the new documents you add to the source document
    leftKeys: ["product.manufacturer_id"], //This field accepts many keys, which amounts to a composite key
    rightKeys: ["_id"],
    //unfortunately, as of now, you can only add subdocuments at the root level, not to arrays of subdocuments
    newKey: "manufacturer" //The upside is that this serve the majority of cases
  })
  //Call exec to run the compiled query and catch any errors and results, in the callback
  .exec(function (err, items) {
    console.log(items);
  });
```

Dokumentide ühendamise protseduur *MongoDBs* (üks populaarsemaid NoSQL süsteeme), mis kirjeldab, kuidas ühendamist teostada.

Product **JOIN** Manufacturer {product\_id, manufacturer\_id, product\_name, manufacturer\_name}

Kahe relatsioonilise muutuja väärtuse ühendamine *Rel* süsteemis kasutades *TutorialD* keelt. Ütlen süsteemile, mida tahan.

Eelistan **deklaratiivset** keelt, sest tahan olla produktiivsem ja kirjutada lühemat koodi, mida süsteem saaks optimeerida. Mida rohkem koodi, seda rohkem võimalusi vigu teha ja seda rohkem on vaja testida.

# Deklaratiivne vs. imperatiivne keel (2)

```
{
  $lookup:
  {
    from: <collection to join>,
    localField: <field from the input documents>,
    foreignField: <field from the documents of the "from" collection>,
    as: <output array field>
  }
}
```

Alates *MongoDB 3.2* saab seal ühendamisoperatsioonide jaoks kasutada *\$lookup* operaatorit.

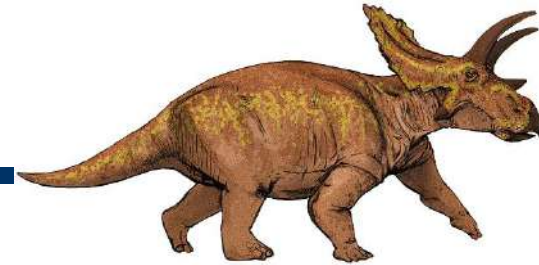
- ♦ Paljudesse NoSQL süsteemidesse on lisandunud kas deklaratiivse andmebaasikeele kasutamise võimalus või uusi operaatoreid, mis võimaldavad panna kirja protseduure rohkem *funktsionaalses* (deklaratiivsemas) stiilis.

# NoSQL süsteemide puudused (2)

- ◆ Muudab raskeks väljaspool konkreetset rakendust (millega seoses NoSQL süsteem on kasutusele võetud) andmetele *ligipääsemise* ning nendest andmetest *arusaamise*.
- ◆ Vähem ranged nõuded transaktsioonidele võivad viia *ebakorreksete* andmete andmebaasi sattumiseni ning sellest tulenevate valede otsusteni.
- ◆ Arengujärgus tehnoloogia; palju konkureerivaid lähenemisi; kerge jääda ühe toote "lõksu".



# SQL on suurepärane kohaneja



- ◆ SQLi on lisandunud/lisandumas võimalused organiseerida andmeid erinevate andmemudelite alusel – luua "**hübriidseid**" andmebaase.
- ◆ SQL-andmebaasis saab hoida, töödelda ja kasutada andmeid, mis on esitatud:
  - JSON/XML dokumentidena,
  - võti-väärtus paaridena (PostgreSQLi *hstore* laiendus),
  - SQL:2020 peaks lisanduma graafiandmete tugi – saab deklareerida, et mingites tabelites olevad andmed on graafid ja teha graafide põhjal päringuid.

NoSQL = Not  
yet SQL??!

# SQL on eeskuju

- ♦ **SQL-laadsed** andmebaasikeeled on leidnud tee paljudesse *NoSQL süsteemidesse*.
  - Cassandra Query Language (Cassandra)
  - N1QL (Couchbase)
  - SPARQL (RDF-põhised andmebaasisüsteemid)
  - PGQL, GQL (graafipõhised süsteemid)
  - SQL++ (AsterixDB, Couchbase Analytics),
  - PartiQL (Amazon Quantum Ledger Database, Couchbase Server)



# SQL on probleemne – näiteid

- ◆ Keeruline, kohmakas, liiga lopsakas süntaks.
- ◆ Keeleline liiasus – ühte ja sama ülesannet saab lahendada (väga) paljudel erinevatel viisidel.
  - Erinevatele lahenduseks olevatele lausetele koostatakse erinevad täitmisplaanid ja sellest tuleneb erinev lause täitmise kiirus.
  - <https://digikogu.taltech.ee/et/item/74411e4c-89b5-4658-b1f7-f4958f8753e0>